고속 Turbo Product 부호 복호 알고리즘 및 구현에 관한 연구

정회원 최 덕 군*, 준회원 이 인 기*, 정회원 정 지 원*

High Speed Turbo Product Code Decoding Algorithm

Duk-Gun Choi*, In-Ki Lee*, Ji-Won Jung* ***Regular Members***

요 약

최근 터보 부호에 비해서 구현시 복잡하지 않고, 높은 부호화율에서 거의 샤논 이론에 접근하는 Turbo Product Code(TPC)에 대해 관심이 고조되고 있다. 본 논문에서는 초고속 통신 시스템에 적용하기 위한 고속 TPC 복호를 위한 세가지의 알고리즘을 제안하는 바이다. 첫째로, 기존의 Turbo Product code 복호기에서 row과 column을 직렬로 복호를 하지 않고 복호 구조가 병렬로 동작하는 Turbo Product code 복호기를 제안한다. 둘째로 반복 중지 알고리즘을 제안하고 마지막으로, P-Parallel 알고리즘을 통해 P rows와 P columns을 병렬로 처리하여 복호한다. 모의 실험을 한 결과 기존의 방식에 비해 복호 지연이 줄어들고 성능면에서 직렬 방식과 거의 비슷한 성능이 나타난다. 또한 고속알고리즘을 바탕으로 VHDL모델링을 하였으며, 이를 timing 시뮬레이션 하여 메모리 요구량 및 복호 속도 향상도를 분석하였다.

Key Words : Turbo Product Codes, Parallel Algorithm, Early Stop, P-Parallel

ABSTRACT

In this paper, we introduce three kinds of simplified high-speed decoding algorithms for turbo product decoder. First, A parallel decoder structure, the row and column decoders operate in parallel, is proposed. Second, HAD(Hard Decision Aided) algorithm is used for early-stopping algorithm. Lastly, P-Parallel TPC decoder is a parallel decoding scheme, processing P rows and P columns in parallel instead of decoding one by one as that in the original scheme.

Ⅰ. 서 론

최근의 무선 통신 시스템은 무선 멀티미디어 전송에 기반을 두고 있기 때문에, 고속 데이터 전송에 효율적이고 성능이 우수한 복호기 개발이 필수적이다. 1993년에 Berrou등에 의해 발표된 Turbo 부호[1]는 Shannon's Limit에 근접한 성능을 나타내지만 많은 데이터양과 연산작용이 매우 복잡해 저속 서비스에만 적용된다. 최근의 오류정정분야의 또 다른 연구는 LDPC(Low Density Parity Check) 부호에 관심이 집중되고 있다. 그러나 LDPC는 복호화가 간단한 반면에 부호화 부분의 높은 복잡도가 LDPC 부호의 최대의 단점이다. 1998년 Pyndiah에 의해 소개된 TPC(Turbo Product Code)[2]는 기존의 LDPC 부호의 단점인 부호화 시 구성 어려움, 그리고 성능 향상을 위한 많은 블록 크기를 요구한다는 것과 Turbo 부호의 많은 계산량과 고속 복호기 구성의 어려움 등의 단점을 보완한 작은 블록 크기를 가로 세로로 product 시킨 후 같은 복잡도로서 많은 블록 크기의 효과를 얻을 수 있고 복호기가 간단하여

* 한국해양대학교 전파공학과 위성통신 연구실(dkchoi@bada.hhu.ac.kr),
논문번호 : KICS2004-11-268, 접수일자 : 2004년 11월 08일
※본 연구는 한국 학술 진흥 재단의 지역대학 우수 과학자 지원 연구사업(과제번호 D00553)의 지원으로 수행되었습니다.

고속 구현이 가능하며, 높은 부호화율에서 Shannon Limit에 근접하는 새로운 차세대 오류정정 부호화 방식으로 무선 멀티미디어 통신을 요구하는 최근의 무선 통신시스템에 오류정정방식으로 적합하다. 그러나 TPC 복호기의 가장 큰 문제점은 두개 복호기가 직렬로 연결된 구조이기 때문에 두개의 복호기가 직렬로 연산 반복하는 과정에서 지연으로 인한 속도 저하를 가져올 수 있다. 따라서 본 논문에서는 초고속 통신 시스템에 적용하기 위한 고속 TPC 복호를 위한 세가지의 알고리즘을 제안하는 바이다. 첫째로, 기존의 Turbo Product code 복호기에서 row과 column을 직렬로 복호를 하지 않고 복호 구조가 병렬로 동작하는 Turbo Product code 복호기를 제안한다. 둘째로 반복 중지 알고리즘을 제안하고 마지막으로, P-Parallel 알고리즘을 통해 P rows와 P columns을 병렬로 처리하여 복호한다. 모의실험을 한 결과 기존의 방식에 비해 복호 지연이 줄어들고 성능면에서 직렬 방식과 거의 비슷한 성능이 나타난다. 또한 고속알고리즘을 바탕으로 VHDL모델링을 하였으며, 이를 timing 시뮬레이션 하여 메모리 요구량 및 복호 속도 향상도를 분석하였다.

Ⅱ. 기존의 Turbo Product Code

TPC 부호는 두개 혹은 그 이상의 짧은 길이의 블록부호 (C_1, C_2)를 이용하여 긴 블록 부호 $(P=C_1\times C_2)$를 만드는 것이 가장 효율적이고 간단한 부호화 알고리즘이다. 그림 1은 TPC 부호화기 구성도를 나타낸다[2].

그림 1과 같이 두 개의 블록 부호를 적용할 경우, k_1(또는 k_2)개의 정보 비트를 가로(또는 세로)로 배치한 후, 가로는 (n_1, k_1, δ_1)을 가지는 블록 코드 C_1으로 부호화 시키고, 세로는 (n_2, k_2, δ_2)를 가지는 C_2로 부호화시켜 전송한다. 따라서 TPC부호 $P=C_1\times C_2$이므로, (n, k, δ)를 가진다. 여기서 $n=n_1\times n_2$, $k=k_1\times k_2$, $\delta=\delta_1\times\delta_2$ 이고 부호화율은 $R=R_1\times R_2$ $(R_1=k_1/n_1, R_2=k_2/n_2)$ 이다. 따라서 두 부호어를 product 함으로써, 높은 부호화율에서 minimum hamming distance의 증가에 의해서 오류정정 능력은 향상된다. 그리고 오류를 산발시키는 효과가 있는 인터리버가 필요치 않으며, 복호시 가로 부분을 먼저 복호 한 후 이를 extrinsic 정보로 이용하여 세로 부분을 복호하면서 반복 복호를 한다. TPC에 적용되는 블록 부호 C_1, C_2는 해밍부호, BCH부호, RS부호등 다양한 블록 부호를 적용시킬 수 있다. Pyndiah가 제안한 TPC복호기 구조는 아래 그림 2와 같다.

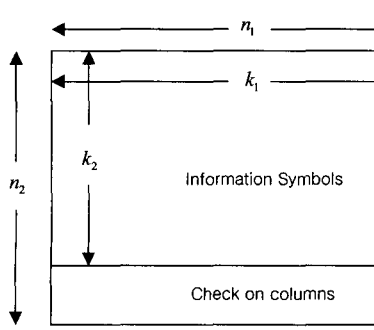


그림 1. TPC 부호화기 구성도 $(P=C_1\times C_2)$

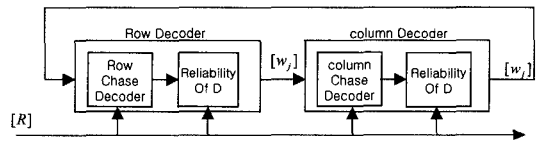


그림 2. TPC 복호기 구조

원 신호 E 그리고 가우시안 잡음 신호 N에 의해 수신신호 벡터 R 다음식과 같이 나타낼 수 있다.

$$R=E+N$$
$$R=(r_1,\cdots, r_l,\cdots r_n)$$
$$E=(e_1,\cdots, e_l,\cdots e_n),$$
$$N=(n_1,\cdots, n_l,\cdots n_n)$$ 이다.

최적 결정 비트 $D=(d_1,\cdots, d_l,\cdots d_n)$는 식 (1)과 같이 maximum likelihood 방식에 의해 결정된다.

$$\begin{aligned} &D=C^i \text{ if } \Pr\{E=C^i|R\}>\Pr\{E=C^j|R\}\forall j\neq i \\ &D=C^i \text{ if } |R-C^i|^2>|R-C^j|^2\forall j\neq i \\ &|R-C^i|^2=\sum_{l=1}^{n}(r_i-c_i^l)^2 \end{aligned} \quad (1)$$

모든 부호어의 집합 C의 i번째 부호어 이다. 이 경우 n값이 크면 계산량이 매우 많고 오래 걸리며 거의 불가능하다. 따라서 1972년 low' complexity 알고리즘을 제안하였으며, 이는 높은 SNR에서 D는 Y의 중심점에서 반경이 $\delta-1$의 구안에 포함될 확률이 높다.

여기서 $Y=(y_1,\cdots, y_l,\cdots y_n)$이며, $y_i=0.5(1+$

$sgn(r_j)$)이다. 후보 가능한 부호어 C를 찾는 chase 알고리즘은 다음과 같다[3].

1단계: $p = \lfloor \delta/2 \rfloor$ 개의 신뢰성 없는 Y의 비트 위치를 수신 벡터 R을 이용해서 결정한다.
신뢰성 없는 비트의 위치는 수신되는
$\Lambda(y_j) = ln(\frac{P_j(e_j=+1/r_j)}{P_j(e_j=-1/r_j)}) = (\frac{2}{\sigma^2}) = \lfloor r_j \rfloor$ 이다.

2단계: q개의 test pattern T^q를 생성한다. T^q의 생성 방법은 n개의 비트 위치 중 $\Lambda(y_j)$가 가장 적은 값에 해당하는 위치 j에 "1"을 위치시키고 나머지 비트위치는 "0"을 삽입하고, $\Lambda(y_j)$가 가장 적은 두개의 비트 위치에 "1"을 위치시키고 나머지 비트는 "0"으로 배치한다. 같은 방법 계속해서 $\Lambda(y_j)$가 가장 작은 p 개의 비트 위치에 "1"을 위치시키고 나머지 비트는 "0"을 삽입한다.

3단계: q개의 T^q를 생성하고 난 뒤에, $Z^q = Y \oplus T^q$하여 오류 위치를 정정한 Z^q를 생성한다.

4단계: Z^q를 블록 복호하여 C^q를 생성한다.

4단계가 완료되면 검토 되어지는 $C^q = (C^1, C^2, C^3, \cdots, C^n)$벡터가 생성 되며, 그림 2의 부호기에서 연판정을 출력하기 위해 D의 j 번째 비트 신뢰도 $(LLR)_j$는 아래 식 (2)와 같다.

$$(LLR)_j = \ln\left(\frac{\Pr(e_j=+1/R)}{\Pr(e_j=+1/R)}\right) \tag{2}$$

식 (2)은 높은 부호율을 가지는 부호어에 대해서는 거의 계산 불가능하고 복잡하므로 Pyndiah 에 의해 식 (3)와 같이 간략하게 최적화 하였다.

$$(LLR)_j \approx \frac{2}{\delta^2}\left(r_j + \sum_{l=1, l\neq j}^{n} r_l\, c_l^{\min(+1)}\, P_l\right)$$
$$P_l = \begin{cases} 0 & if \quad c_l^{\min(+1)} = c_l^{\min(-1)} \\ 1 & if \quad c_l^{\min(+1)} \neq c_l^{\min(-1)} \end{cases} \tag{3}$$

$c_l^{\min(+1)}$ 과 $c_l^{\min(-1)}$은 j 번째 비트가 +1, -1을 가지고 부호어 중 수신신호와 해밍거리가 가장 작은 부호어의 l번째 비트를 나타낸다. 따라서 식(3)는 식 (4)와 같이 나타낼 수 있다.

$$r_j' = r_j + w_j \quad w_j = \sum_{l=1, l\neq j}^{n} r_l\, c_l^{\min(+1)}\, P_l \tag{4}$$

r_j : soft input data , ω_j : extrinsic information

이는 R과 $r_j\,(l\neq j)$의 최소 유클리언 거리의 함수이다. 식 (4)에서 반복 시 생성되는 r_j의 soft decision 값 r_j'는 입력 수신벡터의 r_j와 extrinsic 정보의 합으로서 표시될 수 있다. extrinsic 정보 ω_j는 자기 신호 j번째를 제외한 r_j와 선택된 부호어 C_j의 곱의 합으로 표시 될 수 있다. 즉 r_j'는 chase 알고리즘에 의해 복호된 D의 soft decision 값이다. r_j'는 $r'_j = \beta d^j$ $(\beta \geq 0)$ 와 같이 나타낼 수 있으며, β는 신뢰도 factor 이다. 처음 반복 시에는 신뢰도가 낮으므로 낮은 값으로 반영하면서 반복횟수가 증가 할수록 높게 설정한다. product code를 가로, 세로 복호 시 TPC의 복호 블록도는 아래 그림 3과 같다.

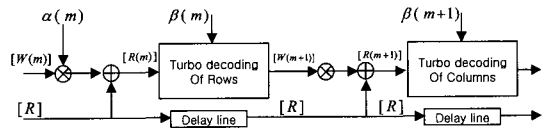


그림 3. Pyndiah가 제안한 TPC 복호기 구조

$$R[2] = [R] + \alpha[2]\cdot[W[2]], \quad W[2] = R[2] - R[1]$$

α는 스케일링(scaling) factor이며, 이는 수신신호 R과 W에 있는 샘플들의 표준편차를 고려한 것이다. 따라서 대부분의 논문에서도 $\alpha[m] = [0,\ 0.2,\ 0.3,\ 0.5,\ 0.7,\ 0.9,\ 1.0,\ 1.0]$, $\beta[m] = [0.2,\ 0.4,\ 0.6,\ 0.8,\ 1.0,\ 1.0,\ 1.0,\ 1.0]$ 으로 할당한다. 다음 그림 4는 위와 같은 방식으로 BCH(63.57.3)code 두개를 사용하여 TPC를 구성하여 각 반복 횟수 따른 성능을 나타낸 것이다.

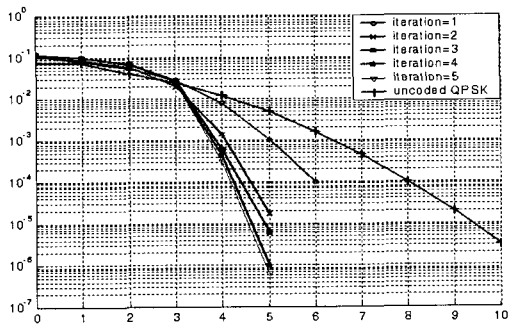


그림 4. BCH(63.57.3) TPC의 반복횟수에 따른 성능

Ⅲ. 본 논문에서 제안하는 고속 복호 알고리즘

3.1 병렬 방식의 복호 알고리즘

앞에서 살펴본 그림 3과 같이 Pyndiah에 의해 제안된 복호기는 세로(또는 가로)를 다음 복호하기 전에 반드시 가로(또는 세로)를 복호 하여야 하는데 반해, 지연을 감소 시키기 위한 본 논문에서 제안된 병렬 TPC복호기 구조는 그림 5와 같다. 그림 5에서와 같이 가로 세로 복호기에 대한 복호는 병렬로 이루어져지며, 복호 지연은 기존의 방식에 비해 절반 효과를 갖기 때문에 복호 속도는 2배 개선 시킨다. 메트릭 $[W^{row}]$와 $[W^{col}]$은 가로와 세로 부호어에 대한 수신 신호의 extrinsic 정보 이며, 각 블록에 대한 복호를 마치고 block-by-block으로 같은 시간에 extrinsic 정보의 교환으로 복호가 수행되어진다. 첫 번째 반복 시, 즉 m=0, 일 시 $[R^{row}(0)]=[R^{col}(0)]=[R]$이라 두고, 가로 세로의 블록 길이는 n이라 가정한다.

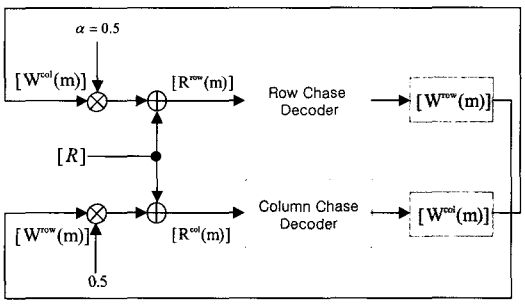


그림 5. 병렬 복호기 구조 및 복호 과정

가로 세로에 대한 복호가 마친후 업데이터된 신호 벡터 $[R^{row}(m+1)]$, $[R^{col}(m+1)]$은 다음 반복을 위한 입력으로 전달 된다. 병렬적으로 업데이터된 신호 벡터는 아래 식 (6)과 같다.

$$[R^{row}(m+1)]=[R]+\alpha(m)\cdot[W^{col}(m)]$$
$$[R^{col}(m+1)]=[R]+\alpha(m)\cdot[W^{row}(m)] \quad (6)$$

3.2 Early-Stop 알고리즘

복호기는 여러 번의 반복횟수에 의해 성능이 개선되어진다. 그러나 어느 일정한 반복횟수가 지나면 에러율은 더 이상 개선되지 않는다(error floor). 이런 경우 더 이상의 반복은 의미가 없으며 이는 복호 속도만 저하시키는 결과를 초래한다. 본 연구에서는 채널 추정된 결과 및 적정한 알고리즘을 바탕으로 더 이상 반복하지 않고 반복을 중지하는 알고리즘을 제안하는 바이다. 직렬 또는 병렬로 연결된 두개의 Row decoder와Column decoder에서 출력된 soft decision 값을 hard decision 하여 복호된 비트값을 비교하여 오류 갯수를 체크한 후에 더 이상 오류를 정정하지 않았으면 더 이상 반복하지 않고 종료하는 알고리즘이다.

3.3 P-Parallel 알고리즘

기존의 TPC 복호기는 row방향과 column 방향으로 N개의 열을 한번씩 복호 함으로써 각각 N번을 복호하게 된다. 본 논문에서 제안하는 P-Parallel TPC 복호기는 P rows와 P column을 병렬로 처리함으로써 N/P번의 복호를 하게 된다. 다음 그림 6 P-Parallel을 이용하여 설계된 TPC decoder이다.

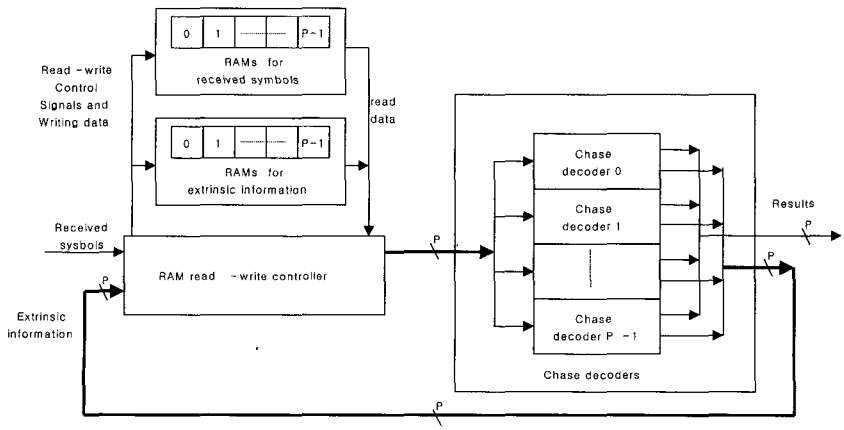
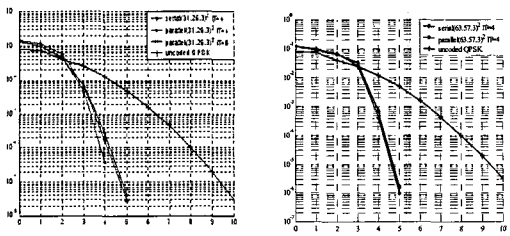


그림 6. P-parallel을 이용한 TPC decoder

3.4 모의실험결과

본 논문에서 모의 실험 시 BCH 부호를 사용하여 구성 하였고 부호화율이 같은 두개의 BCH부호를 사용하였다. Chase 알고리즘을 위해서 가장 신뢰성이 없는 비트를 선택시 $p=4$로 하였다. 그리고 반복 복호를 위한 scaling factor로서 $\alpha=0.5$, $\beta=1$로 고정하였다. β는 모든 반복시 1로 고정하였기 때문에 Chase 복호기의 출력 decoder의 연판정 값을 구하기 위한 β의 곱샘이 필요치 않으며 α는 0.5로 고정했기 때문에 가로 세로에 대한 복호기의 extrinsic 정보의 절반에 해당하는 bit 만큼 우측으로 shift 하면 되기 때문에 H/W구현이 간단해 진다. 이러한 조건을 중심으로 반복 횟수를 4로 하였을 때 결과는 그림 7과 같다. BER=10^{-4}을 기준으로 Pyndiah가 제안한 방식과 비교하였을 때 본 논문에서 제안한 병렬 복호 방식은 0.2dB 이내의 성능감소가 발생된다. 반복횟수가 동일 할 때 기존의 방식과 비교하여 성능의 감소가 일어 나는 것은 첫 번째 반복시 기존의 방식은 row 복호이후 발생된 extrinsic 정보가 column 복호에 제공되지만 본 논문에서 제안한 방식은 column 복호에 제공되는 extrinsic 정보가 0이기 때문이다. 반복 횟수 8번을 같는 parallel(31.26. 3)에서 반복횟수 4를 같는 serial (31.26.3)와 비교시 각각은 같은 지연을 가지지만 병렬방식이 성능이 0.4dB정도 좋아짐을 알 수 있다(그림 7(a)). 표 1에서 보면 알수 있듯이 Early-Stop 알고리즘을 이용하였을 때 E_bN_o가 4dB 일때 약 65%의 복호속도 향상을 가져옴을 확인 할 수 있었다.



(a)BCH(31.26.3)TPC (b)BCH(63.57.3)TPC

그림 7. 기존방식과 병렬 복호방식의 성능비교

표 1. Early-stop 알고리즘을 이용한 평균 반복횟수 (iteration 6회,BCH(31.26.3)TPC)

E_bN_o	평균 반복횟수	복호속도 향상도(%)
1	6	0
2	5.986	0.2
3	4.82	19.7
4	2.049	65.85

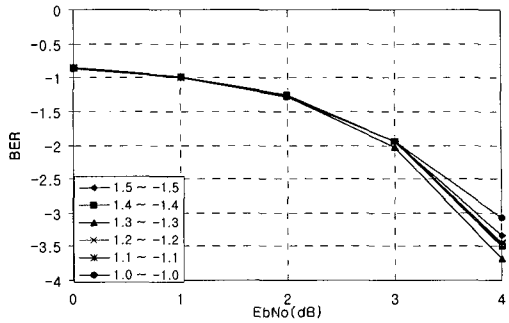


그림 8. 양자화를 위한 수신신호의 최적 범위

Ⅳ. VHDL 모델링

4.1 수신 신호의 양자화 범위에 따른 성능

수신 실수(float) 신호를 양자화를 하기 전에 양자화 하기 위한 신호의 범위를 결정하기 위한 모의 실험을 하였다. 이는 수신신호에 채널 추정값을 곱하는 효과를 나타내기 때문에 양자parallel(31.26.3)을 이용하여 송신된 신호를 수신하였을 때 모두 7bit 양자화를 하여 6개의 범위에 대한 모의 실험을 하였다. 모의 실험 결과 그림 8과 같이 +1.3~-1.3에 대한 범위를 양자화 하였을 때 최적임을 알 수 있다.

4.2 수신 신호의 양자화 비트 수에 따른 성능

앞에서 수신된 BPSK 양자화 구간은 +1.3~-1.3로 결정한 구간에서 양자화 비트 수에 따른 성능을 모의 실험 하였다. 4, 5, 6, 7, 8비트로 각각을 정규화(normalization) 하였고 parallel(31.26.3)을 사용하여 부호화 하였다. 그림 9에서 알 수 있듯이 5비트가 최적의 비트 수 임을 알 수 있다.

수신 신호는 5bit로 양자화 하는 것이 최적임을 알 수 있었다. 수신 신호가 5bit일 경우에 extrinsic 정보는 반복 복호가 진행될 경우 carry가 발생하므로 6bit로 설정하면 된다. 그리고 가로(또는 세로)복

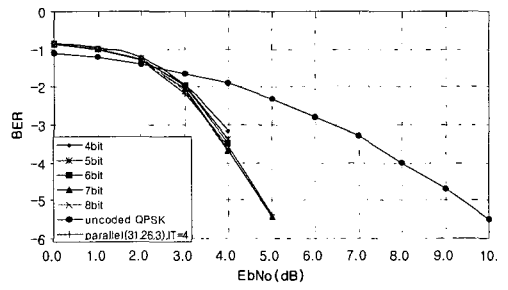


그림 9. 수신신호 비트 수에 따른 TPC 복호기의 성능

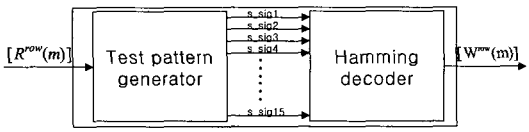


그림 10. chase 복호기의 블록도

호가 끝난 후에 발생된 extrinsic 정보는 scaling factor에 의해서 절반에 해당되는 정보만큼 오른쪽으로 shift시켜서 다시 5bit로 만든 다음 수신신호와 합을 구해서 그 결과가 세로(또는 세로)복호기의 입력이 된다. 본 논문에서 가로(또는 세로)복호기를 VHDL언어로 구성 할 때 hamming(15,11,3)decoder를 사용하였으며 $p=4$하여 복호기 설계하였다. 그림 10은 chase 복호기의 블록도이다.

여기에서 s_sig신호는 chase algorithm에 의해서 발생된 16개 신호와 수신신호의 절대값의 합이다. 이는 하나의 row data에 대해 p=4 이므로 16개의 test pattern이 생성된다.

4.3 test pattern generator 설계

test pattern 발생시 $p=4$이기 때문에 총 16개의 test pattern을 발생 시켜야 한다. 2장에서 설명한 chase algorithm에 의해서 생성 시킨다. 이 방법을 이용할 시 수신된 신호의 가장 신뢰도가 없는 비트에 1을 발생시키고 나머지 비트는 0을 배치하고, 두 번째 신뢰가 없는 비트에 1을 발생시키고 나머지 비트에 0을 배치하면서 16개의 test pattern을 발생해야 한다. 그러나 가장 신뢰도가 작은 순서대로 비트를 찾아서 test pattern을 발생시키는 것이 회로가 복잡해지고 시간이 오래 걸리므로 신뢰도가 없는 비트가 발생하는 순서대로 test pattern을 발생 시켰다.

앞에서 설명한대로 test pattern을 발생시킨 것을 그림 11에서 확인할 수 있다.

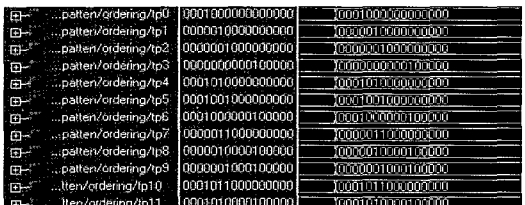


그림 11. test pattern 의 발생

test pattern과 수신 신호의 절대값과의 합인 s_sig는 그림 12와 같이 구해진다.

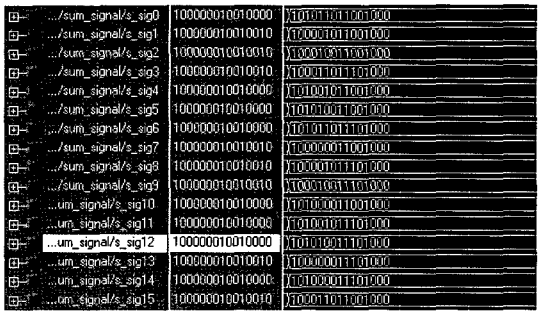


그림 12. 수신신호의 절대치와 test pattern의 합

4.4 hamming decoder 설계

다음 복호에 사용되는 extrinsic 정보를 얻기 위해서 현재 복호된 값과 현재 복호기로의 이전 입력값이 필요하다. 그러기 위해서 복호된 값이 필요하다. 본 논문에서 VHDL로 설계시(15,11,3) hamming decoder를 사용하였다. hamming decoder은 s_sig0~s_sig15를 가지고 복호 한다. 이에 시뮬레이션 결과는 그림 13과 같다. 여기에서 de_bit는 s_sig를 복호한 비트이다.

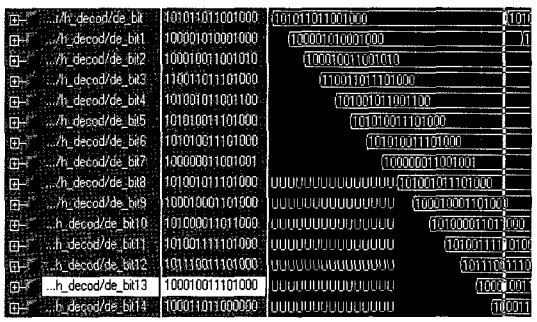


그림 13. hamming decoder의 출력값

다음은 복호된 신호와 수신된 신호와 유클리드 거리를 구해서 가장 복호된 비트들(de_bit~debit14) 중에서 유클리드 거리를 이용해서 가장 작은 값을 선택해서 가로(또는 세로)의 복호 비트로 선택한다. 여기서 유클리드 거리를 구할 때 한 블록을 이루는 값인 15bit에 해당하는 각각의 수신신호와 그에 해당하는 de_bit의 값의 차를 제곱하고 더해서 최소값을 선택하는데 본 논문에서는 제곱을 하지 않고 두 값의 차를 절대치하여 값을 더하였다. 그림 14는 각 디코더에 대한 복호비트와 수신신호 간에 유클리드 거리를 나타내는 그림이다. 여기에서 sum_s는 각 블록에 대한 유클리드 거리를 나타내고 있다.

이렇게 나온 유클리드 거리 중 가장 작은 값을 결정하는 블록도는 그림 15와 같다.

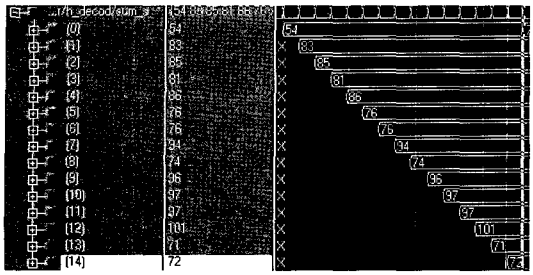


그림 14. 복호된 비트의 유클리드 거리

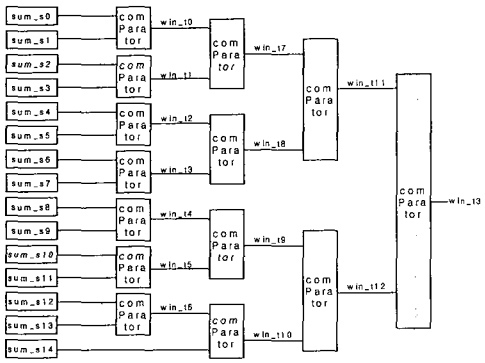


그림 15. 최소 거리 결정 블록도

비교기는 두개의 입력 중에서 위쪽 입력의 유클리드 거리가 작으면 0을 아래쪽에서 작으면 1을 출력한다. 이 출력 값은 win_t에 입력이 되고 최종적으로 win_t13이 나왔을 때 가장 작은 유클리드 거리가 결정된다. 결정 방법은 예를 들어 win_t0=0, win _t7=0, wint_t11=0, wint_t13=0 이면 가장 작은 유클리드 값은 sum_s0가 되고 첫 번째 디코드에서 나온 비트가 최종 결정 비트가 된다.
그림 16은 위에서 언급한 모든 것들을 통합하여 시뮬레이션한 파형이다. 입력신호인 data1과 복호된 신호인 decodebit2가 일치하므로 프로그램의 유효성을 증명할 수 있다.

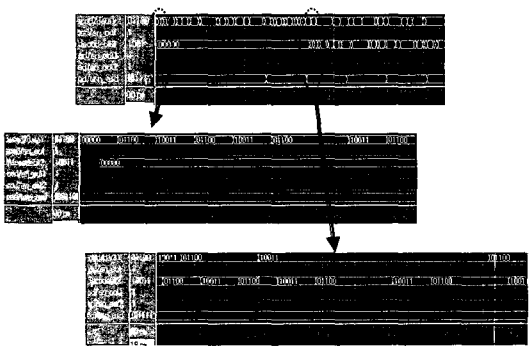


그림 16. 복호된 신호

본 논문에서는 고속화 방안들을 적용하여 VHDL (Very high speed intergreted circuit Hardware Description Languige)코드를 Altera사의 Design complier를 이용 하여 시뮬레이션 하였다.

V. 고속복호기 구현 결과 분석

고속 복호기는 timing 시뮬레이션을 위해 APEX 20KE(EP20K300EQC240-2)칩을 사용하였으며 기존의 직렬 복호기일때와 본 논문에서 제안하는 병렬 고속 복호기를 비교 하였다. 다음 표 2에서 알 수 있듯이 decoding speed 는 P=1일 때 기준으로 기존의 decoder 보다 약 2배의 속도 향상을 가져오고, 요구되는 메모리(Logic cell)은 약 1.5배정도 더 필요함을 알 수 있다. P=2를 기준으로 했을 때 복호 속도는 약 3.6배 향상을 가져오며, 메모리 요구량은 약 2배정도 증가함을 알 수 있다.

표 2. Decoding Speed와 Memory Comparision($BCH\ (15\times11)^2$ iteration=1, Parallel+P-parallel, clock period= 15ns)

	conventional algorithm(P=1)	parallel algorithm	
		P=1	P=2
Memory (Logic cell)	7.024×10^3	10.1×10^3	13.9×10^3
Decoding speed(μs)	23.4	12.7	6.5

본 논문에서 제안하는 복호기의 타이밍 시뮬레이션 결과는 그림 17에 나타내었다. 클럭 주기는 15 [ns]이고 복호방식은 $(15\times11)^2$적용하였다. 5비트로 양자화된 수신 비트르를 가지고 TPC Decoder를 통과한 후 1비트의 복호 비트가 출력이 됨을 알 수 있다.

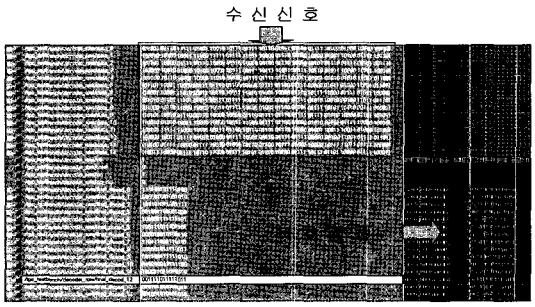


그림 17. TPC Decoder 타이밍 출력도

Ⅵ. 결 론

본 논문에서는 고속 통신을 실현하기 위해서 병렬 TPC 복호기, 반복 중지 알고리즘, P-Parallel 알고리즘을 제안하였다. 기존의 TPC복호기는 row복호 후 column 복호를 하기 때문에 지연으로 인한 복호 속도 저하가 생긴다. 그러나 병렬로 동작하는 복호기는 row와 column이 부분에서 동시에 복호하고 동시에 extrinsic 정보를 넘겨주기 때문에 기존의 방식보다 지연 시간이 작다. BER=10^{-4}을 중심으로 살펴 볼 때 기존의 방식에서 비교 할 때 0.2dB 이내의 성능감소가 일어남을 알 수 있다. 구현 상의 메모리의 요구량은 늘어나지만 기존 방식의 한번 반복을 기준으로 볼 때 같은 수의 메모리가 사용되고 성능이 0.4dB정도 향상됨을 알 수 있다. 그리고 기존의 복호기는 여러번의 반복으로 인해 복호 지연이 생겼으나. 반복 중지 알고리즘을 통해서 약 65%(E_bN_o=4dB)의 복호 속도 향상을 가져왔고, P-Parallel 알고리즘을 통하여 병렬 복호기에서 P=2 일때 기존의 방식에 비해 약 3.6배 정도의 복호속도 향상을 가져온다. 따라서 본 논문에서 제시한 세가지의 알고리즘을 결합하면 초고속 위성통신과 같은 무선 통신에서 고속 복호기의 활용이 가능하다.

참 고 문 헌

[1] Draft ETSI EN 302 307, Digital Video Broadcasting(DVB); Second Generation framing structure, channel coding and modulation for Broadcasting Interactive Services, News Gathering and other broadband satellite applications, 2004. 6.

[2] R. G. Gallager, "Low-Density Parity-Check Codes," IRE trans.information theory, vol.8, pp.21-28,1962.

[3] D. J. C. Mackay and R. M. Neal, "Near Shannon Limit Performance of Low-Density Parity-Check Codes," Electron. Letter, Vol. 32, PP. 1645-1646, Aug.1996.

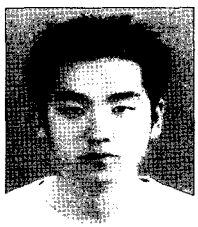
[4] M. Sipser and D. A. Spielman, "Expander Codes," IEEE Trans. Information Theory, vol.42, pp.1720-1722, Aug. 1996

[5] T. Richardson and R. Urbanke, "Effcient Encoding of Low-Density Parity Check Codes," IEEE Trans. Information Theory, vol. 47, pp. 638-656, Feb.2001

[6] J. W. Bond, S. Hui, and H. Schmidt, "Constructing low - density parity - check codes," EURO COMM 2000, Information Systems for Enhanced Public Safety and security. IEEE/AFCEA , pp. 260-262, 2000.

[7] David J. C. Mavkay "Good Error-Correcting Codes Based on Very Sparse Matrics" IEEE Trans. Information Theory, vol. 45, NO. 2. March 1999.

최 덕 군(Duk-Gun Choi) 정회원



2004년 8월 한국해양대학교 전파공학(학사)

2004년 9월~현재 한국해양대학교 전파공학 석사과정

<관심분야> 변 · 복조기술, 채널코딩, FPGA 기술, 위성통신 등

이 인 기(In-Ki Lee) 준회원

2003년 8월 한국해양대학교 공학(학사)

2003년 9월~현재 한국해양대학교 공학석사 과정

<관심분야> 채널 코딩, 변 · 복조기술, FPGA 기술, 위성통신 등

정 지 원(Ji-Won Jung) 정회원

1989년 2월 성균관대학교 전자공학(학사)

1991년 2월 성균관대학교 전자공학(석사)

1995년 2월 성균관대학교 전자공학(박사)

1991년 1월~1992년 2월 LG 정보통신연구소 연구원

1995년 9월~1996년 8월 한국통신 위성통신연구실 선임연구원

1997년 3월~1998년 12월 한국전자통신연구원 초빙연구원

1996년 9월~현재 한국해양대학교 전파공학과 조교수

2001년 8월~2002년 8월 캐나다 NSERC Fellowship(Communication Research Center 근무)

<관심분야> 위성통신, 이동통신, 변 · 복조기술, 채널코딩, FPGA 기술 등